

Evaluating GSD-Aware: A Serious Game for Discovering Global Software Development Challenges

AURORA VIZCAÍNO, FÉLIX GARCÍA, IGNACIO GARCÍA RODRIGUEZ DE GUZMÁN, and M. ÁNGELES MORAGA, Institute of Technologies and Information Systems, University of Castilla-La Mancha

Global Software Development (GSD) is currently a strong industry trend. This means that if computer science engineers are to be trained to deal with this model, it is very important to include the topic in software engineering courses, attempting to ensure that students learn about GSD and become familiar with its advantages and challenges. However, software engineering courses do not always consider including it in their curricula. It must also be recognized that it is difficult to find a suitable method to teach/develop the different skills needed for GSD. There is often a lot of content and not a great deal of time available to teach it. In this article, we propose the use of a serious game called GSD-Aware, with which students can “suffer” some of the typical challenges of GSD by interacting with avatars and by using several means of communication to solve a number of problems posed. The article focuses on the description of the game and on the empirical study conducted to analyze whether GSD-Aware helps students to be conscious of GSD challenges. It was discovered that after 50 minutes playing the game, the students were aware of the greater influence that the following factors can have: lack of coordination, lack of trust, cultural differences, lack of face-to-face and informal communication, time difference, and lack of team spirit. In their final analysis, students agreed that the serious game scenarios helped them to understand what GSD is and to grasp the importance of some GSD challenges.

CCS Concepts: • **Social and Professional Topics** → **Computing Education; Computing Education Programs; Software Engineering Education;**

Additional Key Words and Phrases: Global software engineering, serious game

ACM Reference format:

Aurora Vizcaíno, Félix García, Ignacio García Rodríguez de Guzmán, and M. Ángeles Moraga. 2019. Evaluating GSD-Aware: A Serious Game for Discovering Global Software Development Challenges. *ACM Trans. Comput. Educ.* 19, 2, Article 14 (January 2019), 23 pages.
<https://doi.org/10.1145/3218279>

This work is partially supported by G3SOFT (SBPLY/17/180501/000150) and the TESTIMO project (SBPLY/17/180501/000503) funded by Consejería de Educación, Cultura y Deportes de la Dirección General de Universidades, Investigación e Innovación de la JCCM and GINSENG (TIN2015-70259-C2-1-R) and SEQUOIA (TIN2015-63502-C3-1-R), Ministerio de Economía y Competitividad y Fondo Europeo de Desarrollo Regional FEDER; GLOBALIA (PEII-2014-038-P), Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha (JCCM); G3Soft, SOS and TESTIMO, Consejería de Educación, Cultura y Deportes de la Dirección General de Universidades, Investigación e Innovación de la JCCM.

Authors' Addresses: {Aurora.Vizcaino, Felix.Garcia, Ignacio.GRodriguez, MariaAngeles.Moraga}@uclm.es.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

© 2019 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

1946-6226/2019/01-ART14 \$15.00

<https://doi.org/10.1145/3218279>

1 INTRODUCTION

Current software development has been seriously influenced by globalization, which has led many companies to move from a classical to a geographically distributed development style involving several teams from different countries and cultures. This new development paradigm is called Global Software Development [1]. In addition to providing new opportunities, GSD generates a set of problems that do not typically occur in traditional software development [2]. The challenges of GSD are usually related to communication, coordination, and control, along with issues arising from cultural differences between the different teams [3, 4]. These problems often hinder understanding between project participants, especially when they have to communicate using a common language that is not native to all participants. When team members must use the native language of the other team, misunderstandings may obviously occur that might affect communication, along with the coordination and execution of work, which could represent a risk for the project [2, 4]. Another important aspect that could be affected in such circumstances is trust among the teams involved in a GSD project. Mutual trust is necessary if people are to be able to cooperate and work with each other. A lack of this key commodity may cause the breakdown of coordination efforts between remote teams [5]. This makes it necessary for people working in GSD to possess competencies over and above those required in traditional software development Beecham et al. states in [6].

The teaching of GSD skills at universities is deficient, however, as stated in [7]. In order to determine whether this situation has improved recently, we carried out a study with the objective of detecting how many Spanish universities include Global Software Engineering (GSE) in their curricula.

We obtained a total of 130 curricula, most of which concerned computer science, and others from related fields, such as Bioinformatics, were also analyzed. It was surprising to discover that papers regarding GSD were included as recommended reading for Software Engineering students at only one university. The other curricula studied did not include anything related to this topic. We consider this to be a current issue that should be dealt with; as is stated in [6, 8, 9], educators must prepare tomorrow's software engineers for work in global settings.

Given the nature of the topic, it is difficult to find a suitable method with which to teach these skills, as lectures alone are not sufficient. Having said that, different approaches have been used with this teaching goal in mind, which are described in Beecham et al. [6]. One approach is to conduct distributed courses that involve multiple universities, as explained in [10–17]. In these courses, students located in different countries participate in a software development project with the objective of exposing them to a delocalized situation. This approach would be very useful when there are no time and organizational constraints. Where there are these limitations, which is quite common, this technique is found to be complex to coordinate and time-consuming, as Beecham et al. [6] points out. There is an additional constraint: it is not easy for some universities to find foreign universities with which to collaborate, as they should all have the same time schedule (teaching the subject in the same semester) as well as a similar background and similar motivation.

Another approach is to have students collaborate on projects from companies in the commercial sector. Few software enterprises are willing to do this, however; according to Beecham et al. [6], software enterprises fear that students will not give the project the right priority, or they may believe that students do not have appropriate skills. An intermediate solution is to introduce students to global open-source projects because, as Beecham et al. state in [6], when students participate in these kind of projects “they communicate globally with a team that is experienced in GSD and in integrating new developers” (p. 18). This is a good idea if students can dedicate a lot of time to the project, which is possible if they are studying only the course in GSD. If they have other classes, however, this option could be overly time-consuming.

Finally, in this consideration of ideas on how to teach GSD, one particularly promising approach, described by Beecham et al. in [6], is to use simulators and games focused on GSD processes. An important advantage of this method is that it is not as time-consuming as company project collaboration, since students can use this simulator for several hours in order to simply “get a taste” of the GSD challenges and to research possible solutions to them. Furthermore, using a simulator or a game avoids putting a project at risk, while simultaneously enabling students to be trained in different scenarios and strategies [6, 18].

In earlier work [18], we proposed a serious game that allows students to be aware of the challenges that they might find in GSD. In the previous article, the game was called GSDgame; but we have given it a new name: “GSD-Aware,” as we believe that this provides a better indication of the game’s goal, which is for its users to be aware of GSD challenges. A preliminary evaluation was carried out, which focused mainly on the game quality with reference to a serious game quality model [19]. Positive results were obtained from this evaluation, providing us with some ideas on how it could be improved. For instance, providing more information related to the rewards given by the game was seen to be advisable. However, an evaluation based on the students’ learning is a key factor in the overall assessment. The lack of this evaluation is tackled in this article.

Keeping in mind all of the issues mentioned earlier, this article describes an empirical study that was conducted in order to evaluate students’ learning and perceptions before and after using GSD-Aware. The structure of this article is as follows. In Section 2, we study the subject of the serious game in depth, providing some examples. The features of GSD-Aware are then briefly explained in Section 3. Section 4 presents the evaluation conducted and the results obtained. Our conclusions and future work are outlined in Section 5.

2 SERIOUS GAMES AS A LEARNING TOOL

Many fields of endeavor require people with proper training who have the knowledge, skills, and abilities required to successfully perform their job. These include fields health education, in which the intention is to reduce critical errors in areas such as surgery [20], policy, emergency management, or engineering. Setting up real-life scenarios in such areas provides the students with a solid generation of knowledge, autonomy in the learning process, and a transfer of acquired skills [21]. Traditional teaching methods are often costly and require a lot of preparation time, which is why Serious Games (SGs) are increasingly being introduced. The term SG was first used by Clark Abt in 1970 (as explained in [22]). SGs are designed in such a way that their main purpose is not to entertain the user but rather to train the user in a certain area [23]. This does not mean that the game cannot be enjoyable but that the entertainment derived from the game is designed first and foremost to educate in such a way that the player’s learning experience becomes fun [24]. Some of the distinctive features of these kinds of games are that they are oriented toward training skills in addition to the understanding of complex processes and the simulation of situations that occur in real life [25] or with the goal of making users aware of something [26].

Focusing on the positive effect that SGs have in the knowledge acquisition process, Mettler and Pinto point out in [27] that one of the most important strengths of SGs lies in the fact that they engage students, making them an essential part of the learning and communication process. Furthermore, the educational paradigm is currently undergoing a major shift, moving from the teacher-centered knowledge transfer to a new concept that focuses on the student. This is an approach that is based on the student’s acquisition of certain competencies that are key to professional development. Recently, these competencies have been incorporated into both compulsory education and higher education [24]. The concept of SGs has, therefore, gained relevance, as it represents a new means of acquiring these skills [28]. SGs can be applied to all levels of education, inside and outside the classroom, from children to seniors, and in a wide variety of areas [21]. The

potential for SGs to be a learning tool has been widely-recognized given their ability to balance the entertainment, interactivity, and replayability of typical games with the aim of attaining a particular educational objective [22]. Moreover, the SG approach focuses on learning as a difficult yet rewarding challenge, all in an effort to increase the players' commitment. In the field of Software Engineering teaching, SGs have been used as a support in several stages of the software life cycle and its management. For example:

- In [29], Software Engineering Simulation by Animated Models (SESAM) is presented. In this game, the student plays the role of the project manager and controls the project simulation by using a textual interface.
- SimSE is a computer-based environment that allows the creation and simulation of software-development processes [30].
- In [31], Shaw and Dermoudy describe a simulation game for two software development life cycles. The students who used it successfully identified the reasons why their simulated projects had failed or succeeded. When failure occurred, they determined a strategy to avoid that situation in the future.
- SCRUMIA [32] covers areas such as software project management, in which students (or practitioners in industry) are trained in agile project management techniques by applying SCRUM.
- DELIVER! [33] and SimVBSE [34] are intended to teach and train the concept of Earned Value Management.
- In [35], the authors present an SG with which to teach software project management but includes the possibility of redesigning and creating training scenarios to deal with new skills not considered at first glance.
- "Secret ninja testing with HALO" [36] is intended to develop positive habits in students regarding software testing.
- The authors of [37] describe a game-based environment in which to improve skills related to software security by using simulated real scenarios.

Upon searching the literature regarding the software-development process in GSD in particular, we found it only in the work presented by Noll et al. [38]. In this work, the researchers describe an SG that simulates a GSD environment. The game's objective is to provide students with the expertise required to address the problems that often arise in global development environments. We took the idea of using a budget and a time limit from this game. In our case, however, the students interact with avatars using different means of communication and solve problems in such a way that they become aware of GSD challenges. Our game also helps students reflect on possible solutions to these challenges, which are mainly in terms of cultural, time, and linguistic differences [3, 39–41]. We also consider the problem of lack of trust, as engineers complain that they often feel isolated and that there is a lack of team spirit [42].

3 GSD-AWARE: A SERIOUS GAME FOR GSD

GSD-Aware is a serious game that was designed with the goal of helping Computer Science students to be aware of some problems that occur in GSD. These students do not normally have any previous knowledge of GSD or, even if they are aware of what GSD stands for, they rarely know what challenges this kind of model involves. The game is based on the simulation of scenarios in which the user will play the role of a project manager. The first thing that the student should do is choose from among different avatars, deciding who will be the members of a project on which they are going to work together. This is similar to games in which you choose the players. The avatars simulate being from different cultures and different backgrounds. Students should thus think

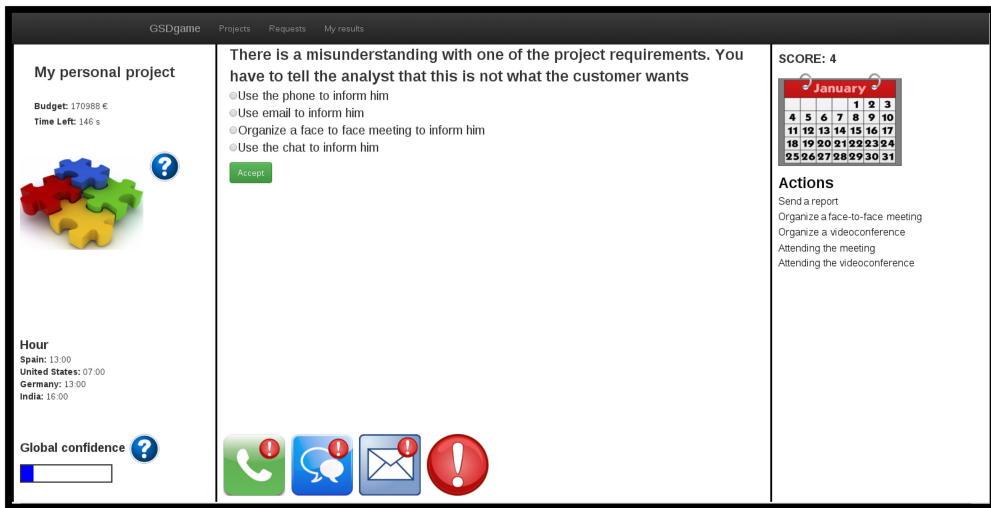


Fig. 1. Screenshots of a scenario in the GSD-Aware SG.

carefully about who is going to work with them on the project. In addition, players are provided with information about the budget and the time that they are going to have available to develop the project, as shown in Figure 1. The main window is divided into three parts:

- On the left-hand side, the information about the project is provided: amount of money available in the budget at any given moment, the time left, and the time in each teammate’s country. In this case, the user is collaborating with simulated colleagues in the United States, Germany, and India.
- Players choose whom they would like to play with from among a selection of avatars; the tools show their nationality and skills. This means that, from the very start, players know what country the other members of their team are from.
- Since confidence or trust is an important issue in GSD, we have also added the Global confidence that this team has. If a student has played with these avatars previously, then the game in this instance gives more value to confidence than the first time that the student plays with these avatars. In the latter case, the confidence is zero.
- The center part of the window shows a scenario with four possible actions to carry out; the student has to choose only one.
- At the bottom of this center part, we can see four icons: the first three are a phone, a chat, and email, all of which contain a red circle with an exclamation mark. This means that the student has a phone call waiting for an answer and that there is also a person chatting to the student. The number of emails in the in-box is also shown. The last large exclamation mark indicates that students have more scenarios to address. The right-hand side shows the calendar on which the individual and shared events are noted in diary format. For example, if the student has to send a report, the relevant date is put on the calendar. Behind this, we can see some of the actions that the student might perform, such as organizing a face-to-face meeting or a videoconference, or simulating attending a meeting or a videoconference organized by another colleague.

Several kinds of scenarios could be shown. For instance, in the scenario shown in Figure 1, the player has four choices. If the player chooses the option of calling the person (option: “use the

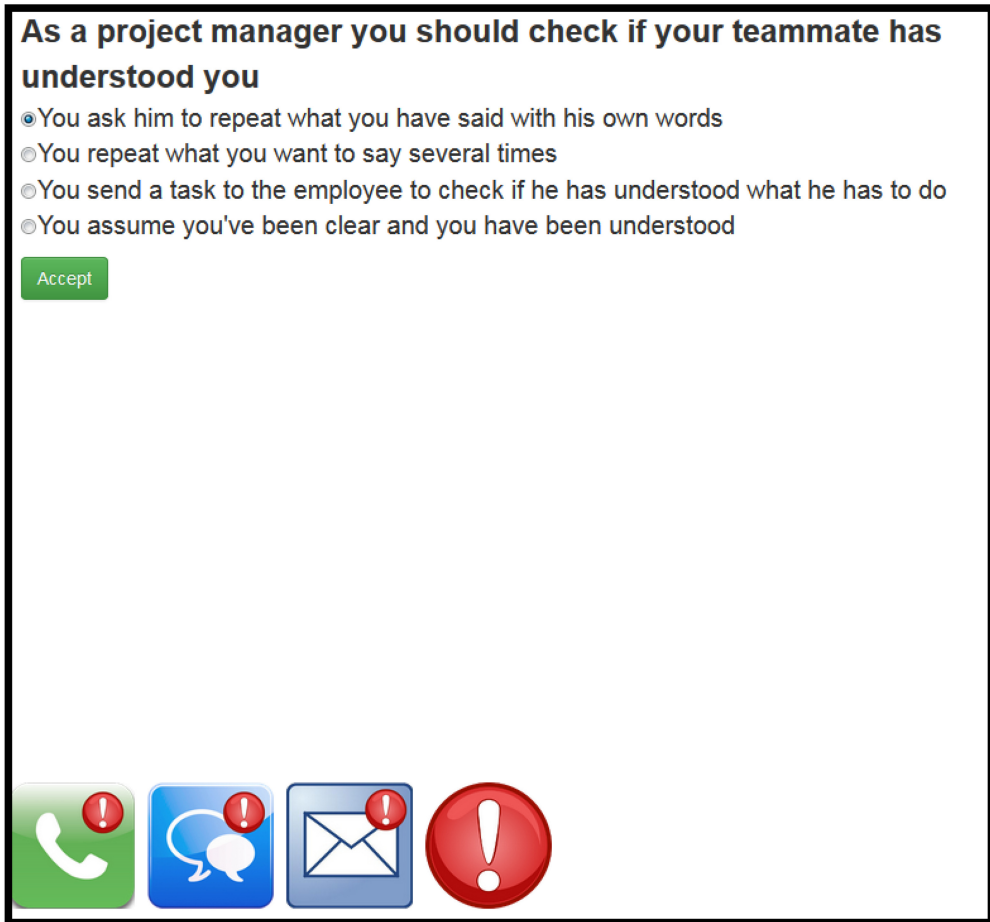


Fig. 2. Zoom of a screenshot showing a cultural factor scenario.

phone to inform him”), the player has to look at the left side of the window to check the time in the country where the other person is working in order to ensure that communication with that person is possible. If this option were chosen and it was nighttime in the other country, the student would lose points, as it is obviously not a good idea to try to contact a person most likely to be sleeping. If the student chooses to meet the person face-to-face (option: “Organize a face-to-face meeting to inform him”), the money assigned by the budget would be depleted, as the student would have to consider the cost of flying to the other country. If the player chooses the best option, which in this case is to email the person, then that student obtains points and gains time. This implies that each alternative has advantages and disadvantages that the student needs to consider. The scenarios focus mainly on problems that arise because of the geographical, cultural, and temporal distance that is present in GSD.

Figure 2 displays a scenario (in this case, only the center part of the window is shown to draw the reader’s attention to the scenario) in which the project manager needs to check on having been understood by the teammate, as there are cultures in which saying “No” is impolite. Given this fact, the project manager should ensure that the other person has truly understood the message. In this case, the best option is the first one; thus, the student wins a point when choosing this.

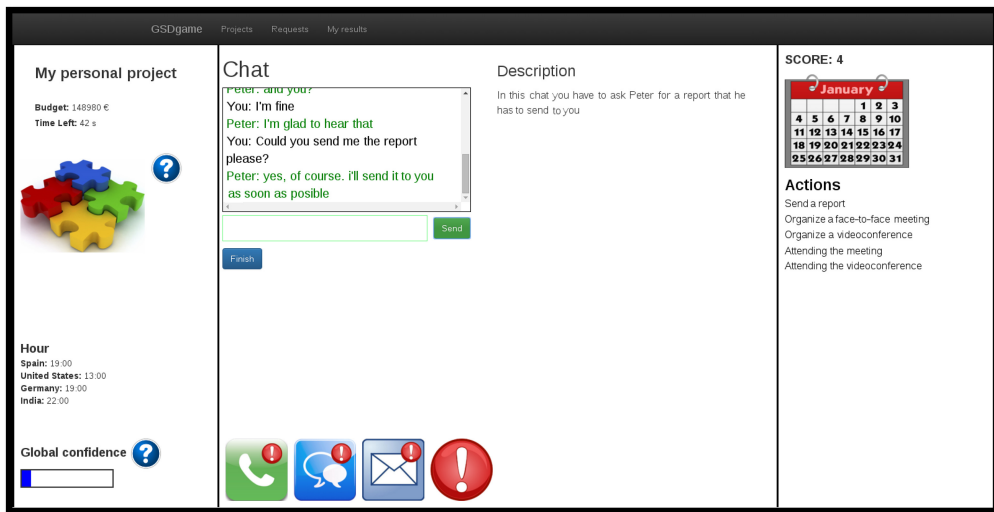


Fig. 3. Complete screenshot of an example of a chat.

At the same time as the student is solving scenarios, the student can receive a simulated phone call, after which the student records the answer with a microphone. The student can listen to the answer before sending it and record it again if dissatisfied with the first version, although time is wasted. The audio is stored in the game when the player presses a respond button that is shown on the screen. The student can also receive emails or chat messages, and when that player presses, for instance, the chat icon, the screen in Figure 3 is displayed. The student can find a description on this screen explaining what the conversation should be about. The student will therefore write some message related to this, and the game will check the sentence by using keywords. When the student writes something in the chat, keywords are then searched for. The game shows different feedback messages, depending on the particular keywords found or missing. For instance, when the student starts a conversation without writing “hello” or “how are you?” or something similar to greet someone and begins the sentence by asking for something, as is the Spanish custom, then the chat shows a feedback sentence similar to the following:

“Feedback: You should have some ice-breaking conversation first. Ask him how he is.”

By checking keywords, we attempt to “teach” the student how to communicate in English in a “neutral” way with people from different nationalities. We should also point out that the chat is checked by using keywords; in the case of the email or phone call, a teacher should look over these when the student has finished the game, informing the student on how to improve the emails or the phone answers by using a face-to-face meeting or via email. The player should always communicate in English. The aim is for the student to “suffer” the challenges of communicating by using a language which is not one’s mother language. In addition, some messages are shown in badly written English or with “false friends” (English words that sound similar to Spanish words but that mean something totally different). The goal is for students to realize the importance of speaking English fluently and understanding it appropriately, since a misunderstanding in a requirement might very well be an important problem. More details about the game can be found in [18].

At the end of a game, a screen is displayed on which the score and the remaining budget are shown. In addition, the student is informed of those questions that the student answered

incorrectly and is provided with brief feedback as to how to avoid these mistakes in the next game. The game does not show the correct answer directly in order to motivate the student to reflect on these mistakes and how they could be solved.

Having described how the game works, the following sections summarize the main characteristics of the game and the pedagogical aspects that have been taken into account during its development.

3.1 Competences

There is currently a growing interest in introducing GSD teaching in the curriculum, and there is some literature describing the skills that practitioners working in global setting should have [43–45]. Most of those skills are related to communication, cultural diversity, team dynamics, time overlap, and working with collaborative technologies. These are the so-called “soft” skills that should be taught in GSD courses [11, 46]. Damian et al. [47] propose a framework for the teaching and assessment of learning GSD skills, describing their findings after using this framework in a course in which the students were geographically distributed.

Taking into account all of the previous proposals and the skills that should be taught and evaluated in GSD learning, we designed scenarios that attempted to cover most of these skills or at least tried to help students realize that they exist and are important when working in GSD. Here, we describe how GSD-Aware covers these skills:

- **Communication.** As mentioned by Paasivaara et al. [11, p. 1129] “Communication is the single biggest challenge of GSD.” Because of this, GSD-Aware shows scenarios with problems that arise owing to a lack of communication in general, a lack of informal communication, or to a lack of face-to-face communication. The student should choose from among different possibilities to solve the problem, as shown in Figures 1 and 2. By making such decisions, the player realizes what could occur when communication is insufficient and how it could be improved.
- **Coordination.** The student faces situations in which work needs to be distributed between team members, with due consideration given to features such as time overlap, members’ profiles, or previous experience working together.
- **Trust.** The tool indicates different scenarios in which there is a problem because of lack of trust. Several possible ways of solving this are shown, such as all of the members spending a weekend together or traveling to visit the colleague. Depending on the project budget, one particular solution could be better than another. The student should choose a particular answer but should be aware that by choosing, for example, the “weekend together” the money available in the budget will decrease more quickly. Both possible answers have the advantage that the time needed to finish the project will be reduced, since the weekend and the visit are supposed to improve mutual trust and ensure that team members will work in a more comfortable environment.
- **Communication tools.** GSD-Aware simulates different communication tools, both synchronous (chat, phone) and asynchronous (email), so that students get experience of communicating in English using these tools. It was curious to see how students got nervous when they had to answer a phone call in English, as most of them were not used to doing so. Another important issue is that students sometimes had problems in communicating in formal emails, as they often communicate informally by email or WhatsApp.
- **Cultural differences.** Typical issues of working with different cultures are shown with different scenarios. For instance, there are some cultures in which it is impolite to say “No.” Therefore, one possible scenario that the student has to face is working with a person from

this kind of culture; the student has to test whether the other person has understood the requirements of a system that is to be implemented. Different solutions are proposed, as shown in Figure 2. In this case, the best solution is the first one; choosing this gives the student points. Other options make the student lose points (2, 1, and 5 points, respectively). The teacher could change the amount of points, adding new options or modifying them.

- **Use of English for communication.** Some false friends and mistakes in the English wording are included in the chats, emails, or phone calls. The aim is to see if the students detect this and show that they realize the problems of communication when nonnative languages are being used. In addition, the student has to record conversations in English and answer the emails and chats in English in order to get used to communicating in this language.
- **Time-zone differences.** The tool shows the time in the different countries (see Figure 1, left-hand side) the aim being for the student to realize, for instance, that it is not possible to communicate using a synchronous means of communication even if the tool proposes it, because it is night in the other person's location.
- **Lack of team spirit.** GSD-Aware shows different situations for dealing with this phenomenon; we could imagine a case in which a member complains of feeling isolated and says that colleagues do not reply to the member's emails. This means that the student, as project manager, has to determine how to manage this situation.

3.2 Features

Beecham et al. [6] describe a set of capabilities that a simulator or game could include. This article was very useful to us, helping us check whether GSD-Aware fulfills these capabilities, as follows:

- **Guides students through the training scenario:** GSD-Aware has different scenarios, focusing mainly on cultural, linguistic, and temporal problems that take place in GSD, which have a variety of levels of difficulty. The student will start with the simplest, with the level of difficulty increasing as the student progresses. The intention is for students to acquire skills gradually.
- **Monitors and assesses students' progress during the simulation:** The game has a points system, which will fluctuate depending on the number of days remaining until the software should be delivered and on the budget available. If the student makes a bad decision when confronting a scenario, this will result in a greater budget loss and a reduction in the days and points remaining; on the other hand, the budgeted money will be safe if the decision has been made correctly.
- **Controls the triggering or emergence of situated or contextual problems, conflicts, and time-zone asymmetries that can arise in GSD projects:** The game shows scenarios with typical problems and failures of GSD, as analysis of real-situation failures allows students to get a grasp of what kind of problems occur [48].
- **Captures and replays simulated GSD process enactments, allowing analysis of the actions taken and their consequences:** Students can play several times and thus try out several solutions, thereby discovering the advantages and disadvantages of each choice.
- **Incorporates GSD tools and repositories:** GSD-Aware can simulate a chat, email, and telephone so that students have to work with both synchronous and asynchronous communication. The application will thus allow random simulation of incoming emails, telephone calls, and chats.
- **Simulates problematic circumstances in GSD, such as mid-project staff termination and budget and schedule reductions:** The student's decisions influence the budget and schedule; wrong decisions mean that the budget is less stable and/or that the

schedule is tighter. Moreover, GSD-Aware provides problematic circumstances; for instance, the student will have an urgent problem that needs a fast solution but when needing to communicate with the colleague in the other site, the student discovers that it is a bank holiday in the colleague's country. The student has to reflect on how to solve this situation.

- **Accommodates student mistakes as learning experiences that are safe and that can simulate differences or gaps in cultural practices:** The tool simulates different cultural and communication issues by means of communication with the avatars. This means that students can be trained without spending time with people in the real world since they learn from the interaction with avatars.

Apart from the features mentioned here, GSD-Aware has a subsystem that can be accessed only by the teacher. This subsystem enables the teacher to create problems, voice calls, chats, projects, see the results of the student's game, and more, as explained in greater detail in [18].

4 EVALUATION OF GSD-AWARE

In this section, the empirical study that was conducted to evaluate the SG is described. In a previous article [18], we described an evaluation of the tool performed by experts in SGs who followed a quality model based on SGs [19]. The results of the evaluation were successful, helping us to improve some features of the tool, mainly those that had to do with how to obtain the rewards. However, one key aspect of the evaluation of the tool was missing, that is, an assessment of whether the students actually learn by using the game.

To achieve this goal, a specific empirical study was planned and conducted by following the recommendations in [49]; this is described next. This empirical study is a quasi-experiment with a pretest-posttest design without a control group [50].

4.1 Participants and Method

The general goal of the study is to determine whether the students learn about GSD challenges. The participants were undergraduate students enrolled in the subject of Software Engineering I (second year). They were chosen because of their suitability; that is, they were students who had sufficient experience in, and knowledge of, software engineering to be able to participate in an experiment about GSD. They had not learned anything about GSD in previous courses; thus, in the pretest, students would answer according to what their intuition or common sense told them.

The students played the game and filled in the surveys in a laboratory class. They spent 50 minutes playing the game only once. The game was in English, although the surveys were in Spanish, as all of the students were Spanish-speakers.

First, the students had to complete an initial survey. This pre-questionnaire, which was designed to establish the baseline, that is, level of knowledge before playing the game, was composed of two parts:

- 2 background questions to evaluate their knowledge of distributed development and GSD, respectively, and one question asking about their previous experience (if any). Both questions were in multiple-choice format, for which there were 5 options, with only one correct answer (the best definition of the term).
- 10 assessment questions (Likert-type scale of 5 values), by means of which the subjects attempted to "guess" the influence of 10 GSD challenges and evaluate them as highly (totally) influential, quite influential, normal, not very influential, or not at all influential. The 10 challenges were chosen after reviewing the GSD literature and analyzing the challenges and recommendations for the design of global software engineering courses, summarized in [8, 51]. The challenges are lack of communication, lack of coordination, lack of trust, lack

of experience, cultural difference, use of English, lack of informal communication, lack of face-to-face communication, time zone difference, and lack of team spirit.

Once the pre-questionnaire was filled in, the students were provided some general instructions about how to play the game, avoiding any type of comment about GSD challenges and any factors that could affect their learning process and the impartiality of the study. In addition, the instructors simulated a scenario and showed subjects the features of the game. The students then spent 50 minutes playing the game in the lab session.

Next, the participants filled in a pre- and post-questionnaire designed to evaluate the learning experience. This questionnaire was composed of the following parts (see Appendix A):

- The same 10 assessment questions as in the pre-questionnaire in order to check whether after playing the game their perception about the influence of the main GSD challenges had changed. The evaluation of these answers allowed us to rate the students' learning experience after using the game.
- 24 questions (Likert-type scale of 5 values) based on the SG evaluation model described in [52].
- 6 questions to rate whether the proposed scenarios in the game helped subjects to understand the importance of the following aspects that might take place in GSD settings: cultural differences, communication and coordination issues, use of another language (English) to communicate, time differences, lack of face-to-face communication, and lack of trust.

To create the questionnaires, the recommendations of [53–55] were followed. First, every effort was made to avoid, or at least minimize, partiality on the part of the individuals conducting the survey by placing special emphasis on suitable wording of the questions. Great care was taken in deciding the questions to be included and in formulating them properly. To reinforce this aspect, the name of each challenge was accompanied by an explanation provided by an example (see Appendix A); this was intended to help those answering the questionnaire to understand each challenge better. Questions were therefore formulated so that respondents could answer them easily and accurately, and the questions were purposeful, specific, and closed. In addition, the number of questions had to be suited to the purpose of our study; we took this aspect into account in our quest to avoid unnecessary questions. A pilot study in which an expert was given the questionnaires was also conducted; and some of that person's recommendations were useful in improving the wording of certain questions. Finally, to mitigate any possible threat related to the material used in this empirical study, it was explained to the subjects during the training session that they would have to fill in the questionnaires provided.

4.2 Results

A total of 40 students took part in the study, which provided the following findings. As far as the knowledge about the concepts of global and distributed developments is concerned, in the preliminary round (pre-questionnaire) 18 subjects chose the correct definition in the question about distributed software development (DSD), and 18 subjects also chose the correct one in the question about GSD. In the round that followed the application of the SG, the correct definition of DSD was chosen by 24 subjects and the correct definition of GSD was chosen by 28 students. Similarly, an analysis of the progress of the students regarding how well they answered questions 1 and 2 before and then after the game showed that almost half the students improved their scores. Thirteen of the students maintained the same score of correct options, and 7 of those studied performed worse in the second round.

Table 1. Frequencies of the Subjects' Perception of GSD Factors: Pre- versus Post-questionnaire

Factor	Round	5	4	3	2	1
Lack of Communication	PRE	25	14	0	0	1
	POST	30	9	1	0	0
	Diff	+5	-5	+1	0	-1
Lack of Coordination	PRE	18	19	2	0	1
	POST	25	14	1	0	0
	Diff	+7	-5	-1	0	-1
Lack of Trust	PRE	4	15	17	3	1
	POST	9	19	10	2	0
	Diff	+5	+4	-7	-1	-1
Lack of Experience	PRE	4	17	13	6	0
	POST	12	12	15	1	0
	Diff	+8	-5	+2	-5	0
Cultural Difference	PRE	0	6	8	16	10
	POST	6	18	10	6	0
	Diff	+6	+12	+2	-10	-10
Use of English	PRE	11	11	10	8	0
	POST	16	13	8	3	0
	Diff	+5	+2	-2	-5	0
Lack of Informal communication	PRE	0	6	19	8	7
	POST	7	13	9	10	1
	Diff	+7	+7	-10	+2	-6
Lack of Face to Face	PRE	5	6	13	10	6
	POST	10	12	8	8	2
	Diff	+5	+6	-5	-2	-4
Time Zone Difference	PRE	3	13	13	8	3
	POST	11	12	12	5	0
	Diff	+8	-1	-1	-3	-3
Lack of Team Spirit	PRE	4	9	16	7	4
	POST	12	14	12	2	0
	Diff	+8	+5	-4	-5	-4

4.2.1 *Perception of GSD Challenges: Descriptive Statistics.* With regard to the perceptions on the part of the students of the 10 challenges that affect GSD, the frequencies of the answers are shown in Table 1.

As can be observed in Table 1, the number of highest-scale value responses (5—It is the totally determining influence) increased for all of the challenges. In most of the challenges, the number of answers in the scales 1, 2, and 3 decreased in the post round, in favor of responses registered in scales 4 and 5. By observing the frequencies of the answers, therefore, a positive trend is identified towards an increase in the number of responses with values 4 and 5. This suggests that subjects were more aware of the importance of the different challenges after playing the game.

Going into greater depth in the analysis of results, Table 2 shows the descriptive statistics, which are complemented by the box plot diagrams displayed in Figure 4.

As may be observed in Table 2 and Figure 4, the general awareness and perception of the students before playing was above the median value (3), except for the cultural difference factor, which had

Table 2. Descriptive Statistics of Subjects' Perception About GSD Factors: Pre vs Post Questionnaire

PRE										
	Lack of Communication	Lack of Coordination	Lack of Trust	Lack of Experience	Cultural Difference	Use of English	Lack of Informal Communication	Lack of Face-to-Face	Time Zone Difference	Lack of Team Spirit
Median	5.00	4.00	3.00	4.00	2.00	4.00	3.00	3.00	3.00	3.00
Std Dev	.749	.797	.876	.877	1.006	1.102	.955	1.231	1.067	1.108
Min	1	1	1	2	1	2	1	1	1	1
Max	5	5	5	5	4	5	4	5	5	5
POST										
Median	5.00	5.00	4.00	4.00	4.00	4.00	3.50	4.00	4.00	4.00
Std Dev	.506	.545	.822	.883	.928	.959	1.125	1.219	1.012	.900
Min	3	3	2	2	2	2	1	1	2	2
Max	5	5	5	5	5	5	5	5	5	5
5. It is the totally determining influence / 4. It has quite an important influence / 3. Normal / 2. It has little influence / 1. It does not influence at all										

a score of 2. This general baseline value indicates that even without previous knowledge of GSD, the students are aware of the importance of certain factors. After playing in the SG session, their awareness and perceptions were the same as before playing with respect to the following factors: lack of communication (5 pre and post); lack of experience in using synchronous and asynchronous communication tools (4); and the use of English for communication (4). The fact that the results did not change in the post-questionnaire is understandable since the students were aware of the significance of these three factors from the very beginning.

Nonetheless, in the post-questionnaire, the importance that was attached to them increased for the following factors: lack of coordination (from 4 to 5), lack of trust between members of the team (from 3 to 4), cultural differences between team members (from 2 to 4), lack of informal communication (from 3 to 3.5), differences in time-zones (from 3 to 4), and a lack of team spirit (from 3 to 4). This provides evidence that the game helped the students to be aware of certain challenges that have to be borne in mind when working in GSD. Especially worthy of note is the perception of how challenging it is to work with people from other cultures. The students had not seen this issue as important in the initial stages. However, it is, in fact, one of the main sources of problems according to GSD literature.

In relation to dispersion of the results, it can be observed that the range and standard deviation of factor ranks was higher before playing the game. Six factors had a range of 4, and five factors had a standard deviation higher than 1, and then in descending order the following: lack of face-to-face communication, lack of team spirit, use of English, time zone differences, and cultural difference. The post-round obtained higher rank difference (std. dev. higher than 1) for the factors lack of face-to-face, lack of informal communication, and time zone differences. In general, all of the factors, except for the factor of lack of informal communication, decreased their dispersion in the post-round, and there was a slight increment in the lack of experience factor. These results suggest that using the game helped to produce a more predictable awareness of the importance of factors on the part of the participants.

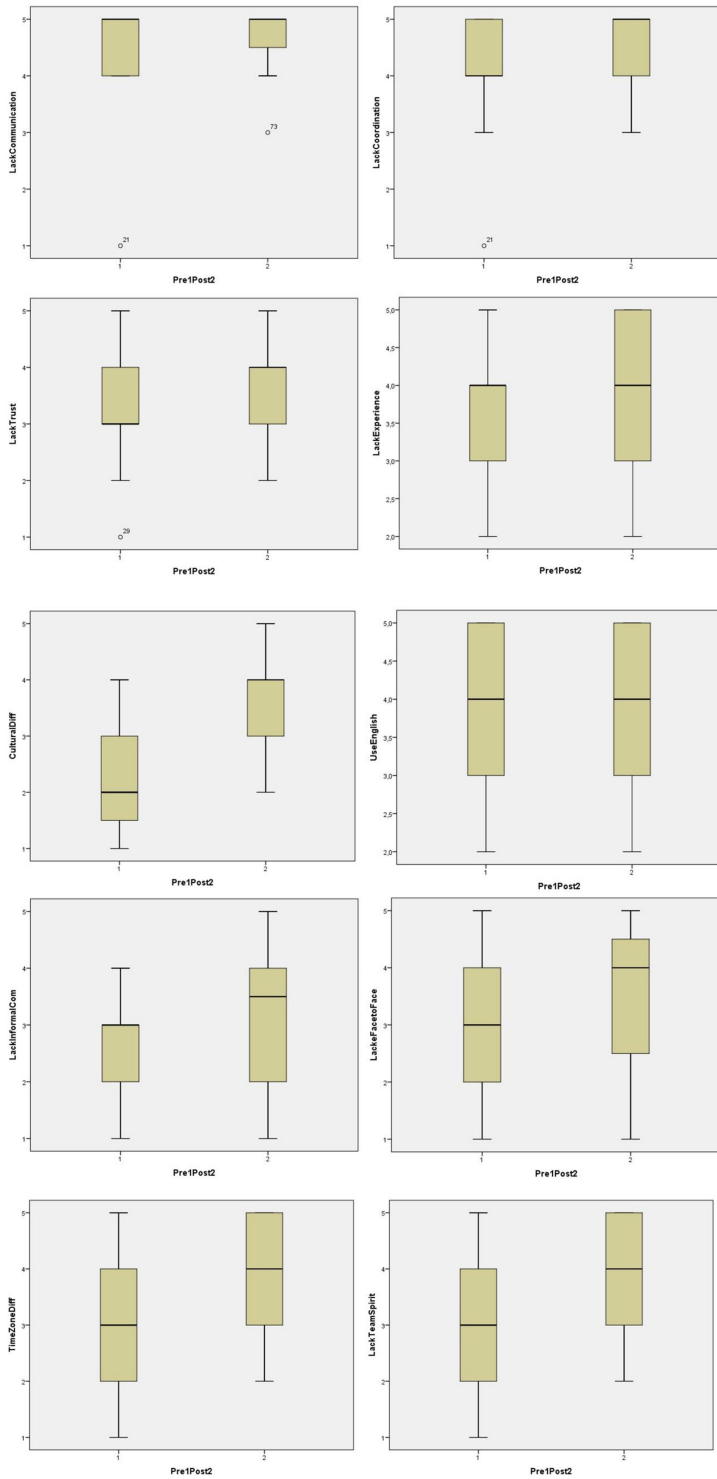


Fig. 4. Box plot diagrams of assessment of challenges before and after students played the SG.

Table 3. Hypothesis Contrast: Pre- versus Post-differences in Subjects' Perceptions of GSD Factors

Factor	Lack of Communication	Lack of Coordination	Lack of Trust	Lack of Experience	Cultural Difference	Use of English	Lack of Informal Communication	Lack of Face-to-Face	Time Zone Difference	Lack of Team Spirit
Kruskal-Wallis (sig)	0.243	0.097	0.029	0.074	0.000	0.082	0.003	0.022	0.019	0.001

4.2.2 Perception of GSD Challenges: Hypotheses Testing. Up to now, we have been able to conclude by observing the results that there were differences in the results obtained in the pre- and post-round. This confirmed that students were more aware about the importance of many of the challenges after playing the game. However, it is essential to determine whether these differences are actually significant.

To check whether there were significant differences in the results in both rounds, a hypothesis contrast was performed, with the following being taken as the null hypothesis for the different factors (from $N = 1$ to $N = 10$).

H_0 : There is no difference between the perception of the students before and after playing the serious game as regards the importance of the N challenge in GSD.

Taking into account that this is a comparison of groups with ordinal variables (an ordered scale of values) and that the dataset of each factor ranks did not follow a normal distribution (the Kolmogorov-Smirnov test was applied), it was decided to use the Kruskal-Wallis statistical test, with a level of significance of 0.05. The results obtained are presented in Table 3.

As can be seen in Table 3, the null hypothesis may be rejected for the factors of lack of trust, cultural difference, a lack of informal, face-to-face communication, time-zone differences and a lack of team spirit. In other words, the perception of students was greater regarding these factors after playing the SG, and this increase was statistically significant. There were four factors in which there were no differences on the students' part, confirming the results shown in Figure 4, except regarding the factor of lack of coordination which, though having a higher median in the post-questionnaire, does not show any significant difference between one round and the other.

4.2.3 Evaluation of the Game Quality. Table 4 shows the results obtained from the evaluation of the quality of the game. With respect to the factors dealing specifically with SG quality (questions 1–24 in the post-questionnaire), there is a clearly positive assessment of this on the part of the students (a score of 4 out of 5). This was especially so regarding the usability of the game (it was easy to understand and to begin to use). Other aspects obtained a neutral score; thus, special attention needs to be paid to these in future versions of the game. In particular, there is a need to work on grabbing the attention of the players from the very beginning and on fostering a fuller engagement with the game. The variety of the tasks also needs to be broadened in the quest to create a greater desire to continue playing and to produce a more comprehensive perception of the usefulness of the game for professional development. The limited amount of time that could be given over to the activity may also have affected the scores that were obtained. All of these considerations lead to the conclusion that, in the future, more complex scenarios have to be designed involving tasks that take longer to complete.

Finally, with regard to the questions that have to do with the usefulness of the scenarios presented by the game (statements 25–30 in the post-questionnaire), a median rate of 4 was obtained (label: I agree) for statements 25, 26, 28, and 30. In addition, a median value of 5 (I agree completely) was obtained for statement 27: "The scenarios in the game have helped me to

Table 4. Results of Subjects' Evaluation of Game Quality

Quality Evaluation Question	Median	Std Dev.	Min	Max
1. The game design is attractive.	4	.966	1	5
2. There was something interesting at the beginning of the game that grabbed my attention.	3	1.022	1	5
3. The variety (form, content, or activities) helped me to keep my attention on the game.	4	.957	2	5
4. The game content is relevant to my interests.	3	.790	2	5
5. The way the game works suits my way of learning.	3	1.021	2	5
6. The game content is connected to other knowledge I already had.	4	.966	1	5
7. It was easy to understand the game and start using it as study material.	5	.882	1	5
8. As I passed through the different parts of the game, I felt confident that I was learning.	3	.963	1	5
9. I am satisfied because I know I will have opportunities to use things I learned playing this game in practice.	4	.963	1	5
10. It is thanks to my own personal effort that I managed to advance in the game.	4	.833	2	5
11. I did not think about my day-to-day routine while playing; I was fully concentrating on the game.	4	1.211	1	5
12. I did not notice the time pass while playing; I only noticed the time when I saw the game had already ended.	4	1.292	1	5
13. I felt myself to be more in the game context than in real life, forgetting what was going on around me.		3	1.165	1
14. This game is appropriately challenging for me; the tasks are neither too easy nor too difficult.	4	1.005	1	5
15. The game progresses at an appropriate pace and does not become monotonous—it offers new obstacles, situations, or variations in its tasks.	3	1.229	1	5
16. I had fun with the game	4	.801	2	5
17. When interrupted at the end of the class, I was disappointed that the game was over.	3	.942	1	5
18. I would recommend this game to my colleagues.	3	1.097	1	5
19. I achieved the goals of the game by applying my knowledge.	4	.966	1	5
20. I had positive feelings on the efficiency of this game.	4	.774	2	5
21. I would like to play this game again.	3	1.127	1	5
22. The game contributed to my learning.	4	.942	1	5
23. The experience with the game will contribute to my professional performance in practice	3	.916	2	5
24. The experience with the game was more efficient for my learning than other activities in the course.	4	1.127	1	5
25. The game scenarios helped me to understand how important time-zone differences can be when coordinating work and in my own communication with others.	4	.778	2	5
26. The game scenarios have helped me to understand the importance of considering cultural differences when working in a multicultural group.	4	.706	3	5
27. The game scenarios have helped me to understand how trying to express myself in a language that is not my mother tongue can have an influence on communication and coordination.	5	.863	2	5
28. The game scenarios have helped me to understand the lack of trust that can arise when you have to work with people you have never met in person before.	4	.843	2	5
29. The game scenarios have helped me to understand the difficulties in communication that may come about when there is no face-to-face communication.	4	.885	2	5
30. The game scenarios have helped me to understand the difficulties in coordination that may arise when there is no face-to-face communication.	4	.844	2	5

understand how communicating in a language that is not one's mother tongue can have an impact on communication and coordination." This seems to indicate that the focus of the scenarios represented the different challenges suitably.

4.3 Limitations of the Study

In order to mitigate possible threats to the validity of this empirical study, some actions were taken. The differences between subjects, learning effects, fatigue effects, persistence effects, subject motivation, and dropout were considered in the planning phase of this study. With respect to differences between subjects, all subjects chosen had a similar knowledge of software engineering, because all of them belong to the same degree course. Learning effects were avoided, as they played only once, a factor that also helped to mitigate possible fatigue effects. Finally, "experimental mortality" refers to the subjects who did not complete the exercises correctly and whose results should be discarded. This did not occur in our case, as all of the subjects finished their assignments and gave us their questionnaires completely filled in.

The following limitations of the study point to the need for future empirical studies in the classroom: The sample size was 40 subjects, which was enough to draw statistically correct conclusions based on the measurements, but replicas with larger sample sizes could produce better conclusion validity. In addition, students with different backgrounds could be considered in an attempt to evaluate the effect of the SG depending on the particular type of student profile.

Finally, in order to be able to generalize the results properly, we have to highlight that although the scenarios provided attempted to represent a typical GSD project, more complex scenarios could be included in the game. The participation of professionals was outside the scope of the present study, as this particular context has to do with considering the inclusion of GSD topics in computer science degree programs. In this case, students are the most suitable participants; in the future, the suitability of the tool for training professionals could be tested by applying it in industry.

5 CONCLUSIONS AND FUTURE WORK

GSD is a growing trend—we should prepare our students to confront the various challenges that they will have to tackle when they become involved in this kind of project. There are a variety of options regarding training students in GSD: involving the student in international projects that work with foreign students, being part of a company's GSD project, and using simulators or SGs. The literature discusses the advantages and disadvantages of these approaches, as described in Beecham et al. [6]. In our case, the first two options were not easy to perform since it is almost impossible to involve students in GSD companies, which do not wish to run any risks. Furthermore, owing to the time constraints of the curriculum, it is not possible to immerse our students in a project with a foreign university. We therefore designed an SG whose goal was to introduce students to GSD challenges.

Having designed this SG, a key step was to test how useful it was. This article describes the evaluation of the game that we designed. The results that we obtained have encouraged us to feel optimistic, as the students' knowledge regarding the concepts of DSD and GSD did indeed improve. Moreover, after playing the game, the students realized that the following concepts have a greater influence than they had initially thought: lack of coordination, lack of trust between members of the team, cultural differences between team members, lack of informal communication, differences in time-zones, and a lack of team spirit.

When the quality of the SG was evaluated, it was concluded that the students had provided a very positive assessment of its usability. Having said that, some features of the game must be analyzed so that they can be enhanced in the future. This is particularly the case in regard to gaining the attention of the players from the very beginning, improving their engagement and

providing players with a greater variety of tasks. More complex scenarios could also be designed. This is easy to do, as the game allows teachers to edit previous scenarios or to add new ones. This leads us to wish to add more complex scenarios to GSD-Aware in our effort to provide our students with new challenges. If we were to do this, students enrolled on the master's degree courses could also use this SG in the Project Management category.

Another proposal for improvement that was suggested by one of the students was for the avatar pictures to be of different races. In the current version, most of the avatars look very European in appearance. Improving the initial interfaces to make them more attractive was another suggestion we received; for instance, instead of showing a picture of a phone, it might be possible to display a phone that is moving and ringing or to show an envelope flying in when a new email appears. These are only ideas at present; nevertheless, they should be discussed with colleagues who are experts in human-computer interaction and graphic user interfaces.

The tool could also be useful for practitioners who are going to become involved in GSD projects, as they would be able to use the tool for practice. They could prepare themselves to meet the new challenges that are involved in GSD but without running any risk from a company's point of view. In an effort to show a greater variety of scenarios, we would also like to take an in-depth look at the Global Teaming Model [56], which contains recommendations for the management of software engineering projects. Our aim would be to see whether scenarios that include some of these recommendations could be added. We may also go on to implement a new version of the tool that would focus more specifically on practitioners. Before doing that, we would like to collect scenarios in which Spanish people often have problems when working in a GSD setting with others from different companies, after which we would extend the scenarios to users who are of other nationalities.

APPENDIX

A EXPERIMENTAL MATERIAL: QUESTIONNAIRES

Post-test Assessment of the Serious Game

Name:

Your comments will help us to improve the game. Thank you!!!

- 1. Please tell us the score you obtained in the game:
- 2. How would you define Distributed Software Development?
 - Software that will be distributed among different computers.
 - Software development carried out by individuals that are distant from each other geographically.
 - Software that is developed in a co-located way.
 - None of the above.
 - All of the above.
- 3. How would you define Global Software Development?
 - As software that is developed for use globally.
 - As a type of paradigm of software development in which the members of the software development team may be located in different countries.
 - A paradigm of development in which partial developments are being integrated into a global product.
 - None of the above.
 - All of the above.

Indicate the level of influence each of these factors has in the success of a global software development project.

It is the totally determining influence	It has quite an important influence	Normal	It has little influence	It does not influence at all				
TI	QI	N	LI	NI				
			TI	QI	N	LI	NI	
Lack of communication For example: Your colleague doesn't answer your emails, etc.								
Lack of coordination For example: Your colleague does not tell you that since he or she has now finished the module, you have to begin yours.								
Lack of trust between team members For example: You don't trust how the other person will debug your code, because you have never seen the individual in person.								
Lack of experience in the use of synchronous and asynchronous communication For example: Not knowing whether it is better to use one tool rather than another, or the tool is used very inefficiently.								
Cultural differences between members of the team For example: There are cultures in which you do not speak to your boss unless the boss asks you to.								
Use of English for communication For example: Misunderstandings caused by the misuse of words known as "false friends," such as "actually."								
Lack of informal communication For example: Work is the only topic of conversation.								
Lack of face-to-face communication For example: You do not know some of the members of the work group personally.								
Time-zone differences: For example: Your working hours do not completely coincide with those of your project colleagues.								
Lack of team spirit For example: You feel isolated working in a distributed way, not being able to share an office with all the members of the team.								

Score: Totally agree, agree, neither agree nor disagree, don't agree, totally disagree.

Totally agree	Agree	Neither agree nor disagree	Disagree	Totally disagree				
TA	A	NN	D	TD				
			TA	A	NN	D	TD	
1. The game design is attractive.								
2. There was something interesting at the beginning of the game that captured my attention.								
3. The variety (form, content, or activities) helped me to keep attention on the game.								
4. The game content is relevant to my interests.								
5. The way the game works suits my way of learning.								

(Continued)

Totally agree	Agree	Neither agree nor disagree	Disagree	Totally disagree
TA	A	NN	D	TD
6. The game content is connected to other knowledge I already had.				
7. It was easy to understand the game and start using it as study material.				
8. As I passed through the different parts of the game, I felt confident that I was learning.				
9. I am satisfied because I know I will have opportunities to use things I learned playing this game in practice.				
10. It is thanks to my own personal effort that I managed to advance in the game.				
11. I did not think about my day-to-day routine while playing; I was fully concentrating on the game.				
12. I did not notice the time pass while playing; I only noticed the time when I saw the game had already ended.				
13. I felt myself more in the game context than in real life, forgetting what was going on around me.				
14. This game is appropriately challenging for me; the tasks are neither too easy nor too difficult.				
15. The game progresses at an appropriate pace and does not become monotonous—it offers new obstacles, situations. or variations in its tasks.				
16. I had fun with the game.				
17. When interrupted at the end of the class, I was disappointed that the game was over.				
18. I would recommend this game to my colleagues.				
19. I achieved the goals of the game by applying my knowledge.				
20. I had positive feelings on the efficiency of this game.				
21. I would like to play this game again.				
22. The game contributed to my learning.				
23. The experience with the game will contribute to my professional performance in practice.				
24. The experience with the game was more efficient for my learning than other activities in the course.				
25. The game scenarios helped me to understand how important time-zone differences can be when coordinating work and in my own communication with others.				
26. The game scenarios have helped me to understand the importance of considering cultural differences when working in a multicultural group.				
27. The game scenarios have helped me to understand how trying to express myself in a language that is not my mother tongue can have an influence on communication and coordination.				
28. The game scenarios have helped me to understand the lack of trust that can arise when you have to work with people you have never met in person before.				
29. The game scenarios have helped me to understand the difficulties in communication that may come about when there is no face-to-face communication.				
30. The game scenarios have helped me to understand the difficulties in coordination that may arise when there is no face-to-face communication.				

ACKNOWLEDGMENTS

We are very grateful to David Valencia for implementing the tool.

REFERENCES

- [1] J. D. Herbsleb and D. Moitra. 2001. Global software development. *IEEE Software* 18, 2 (2001), 16–20.
- [2] C. Ebert, M. Kuhmann, and R. Prikladnicki. 2016. Global software engineering: An industry perspective. *IEEE Software* 33, 1 (2016), 105–108.

- [3] A. Vizcaíno, F. García, M. Piattini, and S. Beecham. 2016. A validated ontology for global software development. *Computer Standards and Interfaces* (2016), 66–78.
- [4] M. Monasor, M. Piattini, and A. Vizcaíno. 2009. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering* (2009), 1–16.
- [5] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt. 2010. Empirical evidence in global software engineering: A systematic review. *Empirical Software Engineering* 15, 1 (2010), 91–118.
- [6] S. Beecham, T. Clear, D. Damian, J. Barr, J. Noll, and W. Scacchi. 2017. How best to teach global software engineering? Educators are divided. *IEEE Software* 34, 1 (2017), 16–19.
- [7] F. Matthes, C. Neubert, C. Schulz, C. Lescher, J. Contreras, R. Laurini, B. Rumpler, D. Sol, and K. Warendorf. 2011. Teaching global software engineering and international project management. In *Proceedings of the 3rd International Conference on Computer Supported Education*. Noordwijkerhout, the Netherlands, 5–15.
- [8] T. Clear, S. Beecham, J. Barr, M. Daniels, R. McDermott, M. Oudshoorn, A. Savickaite, and J. Noll. 2015. Challenges and recommendations for the design and conduct of global software engineering courses: A systematic review. *Proceedings of the 2015 ITiCSE on Working Group Reports* 1–39.
- [9] S. Beecham, T. Clear, J. Barr, M. Daniels, M. Oudshoorn, and J. Noll. 2017. Preparing tomorrow’s software engineers for work in a global environment. *IEEE Software* 34, 1 (2017), 9–12.
- [10] F. Peña-Mora, R. Struminger, J. Favela, and R. Losey. 2000. Supporting a project-based, collaborative, distance learning lab. In *Proceedings of the Eighth International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII)*. 170–176.
- [11] M. Paasivaara, C. Lassenius, D. Damian, P. Rätty, and A. Schröter. 2013. Teaching students global software engineering skills using distributed scrum. In *35th International Conference on Software Engineering (ICSE’13)*. IEEE, 1128–1137.
- [12] M. Paasivaara, K. Blincoe, C. Lassenius, D. Damian, J. Sheoran, F. Harrison, P. Chhabra, A. Yussuf, and V. Isotalo. 2015. Learning global agile software engineering using same-site and cross-site teams. In *Proceedings of the 37th International Conference on Software Engineering—Volume 2*. IEEE Press, 285–294.
- [13] M. Nordio, H. C. Estler, B. Meyer, N. Aguirre, R. Prikladnicki, E. Di Nitto, and A. Savidis. 2014. An experiment on teaching coordination in a globally distributed software engineering class. In *27th Conference on Software Engineering Education and Training (CSEET’14)*. IEEE, 109–118.
- [14] M. Kuhrmann and J. Münch. 2016. Distributed software development with one hand tied behind the back: A course unit to experience the role of communication in GSD. In *11th International Conference on Global Software Engineering Workshops (ICGSEW’16)*. IEEE, 25–30.
- [15] J. Kroll, C. Q. Santos, L. S. Machado, S. Marczak, and R. Prikladnicki. 2016. Challenges and lessons learned on preparing graduate students for GSE work: Brazilians perceptions on a multi-site course experience. In *11th International Conference on Global Software Engineering Workshops (ICGSEW)*. IEEE, 37–42.
- [16] O. Gotel, V. Kulkarni, M. Say, C. Scharff, and T. Sunetnanta. 2009. A global and competition-based model for fostering technical and soft skills in software engineering education. In *22nd Conference on Software Engineering Education and Training (CSEET’09)*. IEEE, 271–278.
- [17] T. Clear and D. Kassabova. 2008. A course in collaborative computing: Collaborative learning and research with a global perspective. *ACM SIGCSE Bulletin* 40, 1 (2008), 63–67.
- [18] D. Valencia, A. Vizcaíno, L. García-Mundo, M. Piattini, and J. P. Soto. 2016. GSDgame: A serious game for the acquisition of the competencies needed in GSD. In *11th International Conference on Global Software Engineering Workshops (ICGSEW)*. IEEE, 19–24.
- [19] L. García-Mundo, M. Genero, and M. Piattini. 2015. Applying a serious game quality model. In *International Conference on Serious Games, Interaction and Simulation*. Springer, 12–20.
- [20] M. Graafland, J. M. Schraagen, and M. P. Schijven. 2012. Systematic review of serious games for medical education and surgical skills training. *British Journal of Surgery* 99, 10 (2012), 1322–1330.
- [21] H. Mouaheb, A. Fahli, M. Moussetad, and S. Eljamali. 2012. The serious game: What educational benefits?. *Procedia-Social and Behavioral Sciences* 46 (2012), 5502–5508.
- [22] D. Djaouti, J. Alvarez, J. Jessel, and O. Rampnoux. 2011. Origins of serious games. In *Serious Games and Edutainment Applications*, M. Ma, A. Oikonomou, L. Jain (Eds.). Springer, 25–43.
- [23] C. S. Loh, Y. Sheng, and D. Ifenthaler. 2015. Serious games analytics: Theoretical framework. In *Serious Games Analytics*. Springer, 3–29.
- [24] M. Guenaga, S. Arranz, I. Rubio, E. Aguilar, A. Ortiz de Guinea, A. Rayón, M. Bezanilla, and I. Menchaca. 2013. Serious games para el desarrollo de competencias orientadas al empleo. *VAEP-RITA* 1, 35–41.
- [25] R. Koster. 2013. *Theory of Fun for Game Design*. O’Reilly Media, Inc.
- [26] W. Peng, M. Lee, and C. Heeter. 2010. The effects of a serious game on role-taking and willingness to help. *Journal of Communication* 60, 4 (2010), 723–742.

- [27] T. Mettler and R. Pinto. 2015. Serious games as a means for scientific knowledge transfer—A case from engineering management education. *IEEE Transactions on Engineering Management* 62, 2 (2015), 256–265.
- [28] A. De Gloria, F. Bellotti, and R. Berta. 2014. Serious games for education and training. *International Journal of Serious Games* 1, 1, 1–15.
- [29] A. Drappa and J. Ludewig. 2000. Simulation in software engineering training. In *Proceedings of the 2000 International Conference on Software Engineering*. IEEE, 199–208.
- [30] E. O. Navarro, A. Baker, and A. Van Der Hoek. 2004. Teaching software engineering using simulation games. In *Proceedings of the 2004 International Conference on Simulation in Education (ICSIE'04)*.
- [31] K. Shaw and J. Dermoudy. 2005. Engendering an empathy for software engineering. In *Proceedings of the 7th Australasian Conference on Computing Education—Volume 42*. Australian Computer Society, Inc., 135–144.
- [32] C. G. von Wangenheim, R. Savi, and A. F. Borgatto. 2013. SCRUMIA—An educational game for teaching SCRUM in computing courses. *Journal of Systems and Software* 86, 10 (2013), 2675–2687.
- [33] C. G. von Wangenheim, R. Savi, and A. F. Borgatto. 2012. DELIVER!—An educational game for teaching earned value management in computing courses. *Information and Software Technology* 54, 3 (2012), 286–298.
- [34] A. Jain and B. Boehm. 2006. SimVBSE: Developing a game for value-based software engineering. In *Proceedings of the 19th Conference on Software Engineering Education and Training*. IEEE, 103–114.
- [35] A. Calderón, M. Ruiz, and R. V. O'Connor. 2017. ProDecAdmin: A game scenario design tool for software project management training. In *European Conference on Software Process Improvement*. Springer, 241–248.
- [36] J. Bell, S. Sheth, and G. Kaiser. 2011. Secret ninja testing with HALO software engineering. In *Proceedings of the 4th International Workshop on Social Software Engineering*. ACM, 43–47.
- [37] C. Jordan, M. Knapp, D. Mitchell, M. Claypool, and K. Fisler. 2011. CounterMeasures: A game for teaching computer security. In *Proceedings of the 10th Annual Workshop on Network and Systems Support for Games*, Vol. 7. IEEE Press.
- [38] J. Noll, A. Butterfield, K. Farrell, T. Mason, M. McGuire, and R. McKinley. 2014. GSD Sim: A global software development game. In *IEEE International Conference on Global Software Engineering Workshops (ICGSEW'14)*. IEEE, 15–20.
- [39] A. Vizcaino, F. García, J. C. Villar, M. Piattini, and J. Portillo. 2013. Applying Q-methodology to analyse the success factors in GSD. *Information and Software Technology* 55, 7 (2013), 1200–1211.
- [40] H. Holmstrom, E. Ó. Conchúir, J. Agerfalk, and B. Fitzgerald. 2006. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *International Conference on Global Software Engineering (ICGSE'06)*. IEEE, 3–11.
- [41] T. Clear. 2010. Exploring the notion of cultural fit in global virtual collaborations. *ACM Inroads* 1, 3 (2010), 58–65.
- [42] N. Panteli and E. Duncan. 2004. Trust and temporary virtual teams: alternative explanations and dramaturgical relationships. *Information Technology & People* 17, 4 (2004), 423–441.
- [43] K. Swigger, R. Brazile, F. C. Serce, G. Dafoulas, F. N. Alpaslan, and V. Lopez. 2010. The challenges of teaching students how to work in global software teams. In *Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*. IEEE, 1–30.
- [44] I. Richardson, S. Moore, D. Paulish, V. Casey, and D. Zage. 2007. Globalizing software development in the local classroom. In *20th Conference on Software Engineering Education & Training (CSEET'07)*. IEEE, 64–71.
- [45] M. J. Monasor, A. Vizcaino, M. Piattini, and I. Caballero. 2010. Preparing students and engineers for global software development: A systematic review. In *5th IEEE International Conference on Global Software Engineering (ICGSE'10)*. IEEE, 177–186.
- [46] D. Damian, C. Lassenius, M. Paasivaara, A. Borici, and A. Schröter. 2012. Teaching a globally distributed project course using Scrum practices. In *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD'12)*. IEEE, 30–34.
- [47] D. Damian, A. Hadwin, and B. Al-Ani. 2006. Instructional design and assessment strategies for teaching global software development: A framework. In *Proceedings of the 28th International Conference on Software Engineering*. ACM, 685–690.
- [48] L. Bernstein and D. Klappholz. 2001. Getting Software Engineering Into Our Guts. *The Journal of Defense Software Engineering, CrossTalk* 14, 7, 25–26.
- [49] C. Wohlin, P. Runeson, M. Höst, M. Ohlson, B. Regnell, and A. Wesslén. 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers. Dordrecht, the Netherlands.
- [50] N. J. Salkind. 2010. *Encyclopedia of Research Design*. SAGE Publications Ltd, Thousand Oaks, CA.
- [51] S. Beecham, T. Clear, J. Barr, and J. Noll. 2015. Challenges and Recommendations for the Design and Conduct of Global Software Engineering Courses: A Systematic Review Protocol. Technical Report number Lero_TR_2015_01.
- [52] R. Savi, C. Gresse von Wangenheim, and A. F. Borgatto. 2011. Analysis of an Evaluation Model of Educational Games. Technical Report INCOD. National Institute for Research and Technology on Digital Convergence (INCOD)/UFSC, Brazil.

- [53] B. A. Kitchenham and S. L. Pfleeger. 2002. Principles of survey research: Part 3: Constructing a survey instrument. *ACM SIGSOFT Software Engineering Notes* 27, 2 (2002), 20–24.
- [54] B. A. Kitchenham and S. L. Pfleeger. 2002. Principles of survey research: Part 2: Designing a Survey. Software. *ACM SIGSOFT Software Engineering Notes* 27, 1 (2002), 18–20.
- [55] B. A. Kitchenham and S. L. Pfleeger. 2002. Principles of survey research: Part 4: Questionnaire Evaluation. *ACM SIGSOFT Software Engineering Notes* 27, 3 (2002), 20–23.
- [56] I. Richardson, V. Casey, F. Mccaffery, J. Burton, and S. Beecham. 2012. A process framework for global software engineering teams. *Information and Software Technology* 54, 11 (2012), 1175–1191.

Received April 2017; revised March 2018; accepted March 2018